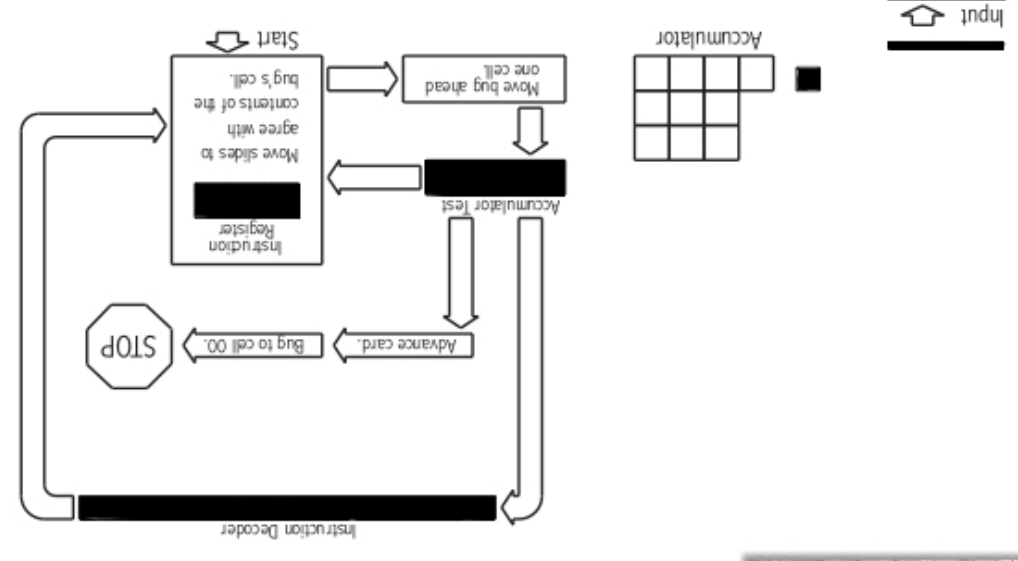
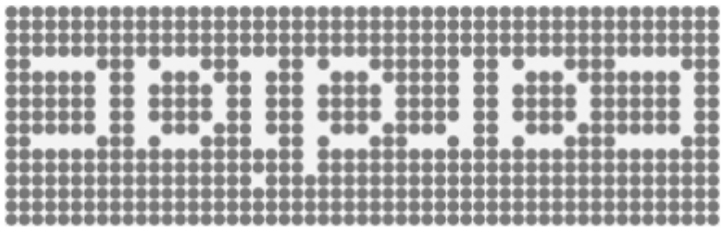
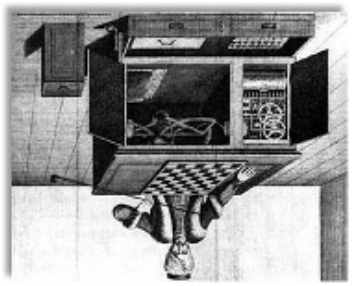






CARDIAC (*CARDboard Illustrative Aid to Computation)* is a learning aid developed by David Hagelbarger and Saul Fingerman for Bell Telephone Laboratories in 1968 to teach high school students how computers work. The kit consisted of an instruction manual and a die-cut cardboard "computer".

The computer "operates" by means of pencil and sliding cards. Any arithmetic is done in the head of the person operating the computer. The computer operates in base 10 and had 100 memory cells which could hold signed numbers from 0 to ±999. It had an instruction set of 10 instructions which allowed CARDIAC to add, subtract, test, shift, input, output and jump.

Opcode	Mnemonic	Instruction	Description
0	INP	**Input**	Take a number from the input card and put it in a specified memory cell.
1	CLA	**Clear & Add**	Clear the accumulator and add the contents of a memory cell to the accumulator.
2	ADD	**Add**	Add the contents of a memory cell to the accumulator.
3	TAC	**Test Accum**	Performs a sign test on the contents of the accumulator; if minus, jump to a specified memory cell.
4	SFT	**Shift**	Shifts the accumulator x places left, then y places right, where x is the upper address digit and y is the lower.
5	OUT	**Output**	Take a number from the specified memory cell & write it on the output card
6	STO	**Store**	Copy the contents of the accumulator into a specified memory cell.
7	SUB	**Subtract**	Subtract the contents of a specified memory cell from the accumulator.
8	JMP	**Jump**	Jump to a specified memory cell. The current cell number is written in cell 99. This allows for one level of subroutines by having the return be the instruction at cell 99 which has '8' hardcoded as the first digit.
9	HRS	**Halt & Reset**	Move bug to the specified cell, then stop program

Op Codes

0 INP Input

1 CLA Clear & Add

2 ADD Add

3 TAC Test Accum

4 SFT Shift

5 OUT Output

6 STO Store

7 SUB Subtract

8 JMP Jump

9 HRS Halt & Reset

Output

Memory Cells

No.	Contents	No.	Contents	No.	Contents	No.	Contents	No.	Contents	No.	Contents
00	001	17		34		51		68		85	
01		18		35		52		69		86	
02		19		36		53		70		87	
03		20		37		54		71		88	
04		21		38		55		72		89	
05		22		39		56		73		90	
06		23		40		57		74		91	
07		24		41		58		75		92	
08		25		42		59		76		93	
09		26		43		60		77		94	
10		27		44		61		78		95	
11		28		45		62		79		96	
12		29		46		63		80		97	
13		30		47		64		81		98	
14		31		48		65		82		99	8__
15		32		49		66		83			
16		33		50		67		84			

Hardware

The “CPU” of the computer consists of 4 slides that move various numbers and arrows to have the flow of the real CPU (the user's brain) move the right way. They have one flag (+/-), affected by the result in the accumulator.

Memory consists of the other half of the cardboard cutout. There are 100 cells. Cell 0 is “ROM”, always containing a numeric "1"; cells 1 to 98 are “RAM”; available for instructions and data; and cell 99 can best be described as “EEPROM”.

Memory cells hold signed decimal numbers from 0 to ±999 and are written with a pencil. Cells are erased with an eraser. A “bug” is provided to act as a program counter, and is placed in a hole beside the current memory cell.

Programming

CARDIAC has a 10 instruction machine language. An instruction is three decimal digits (the sign is ignored) in the form OAA. The first digit is the op code (O); the second and third digits are an address (AA). Addressing is one of accumulator to memory absolute, absolute memory to accumulator, input to absolute memory and absolute memory to output.

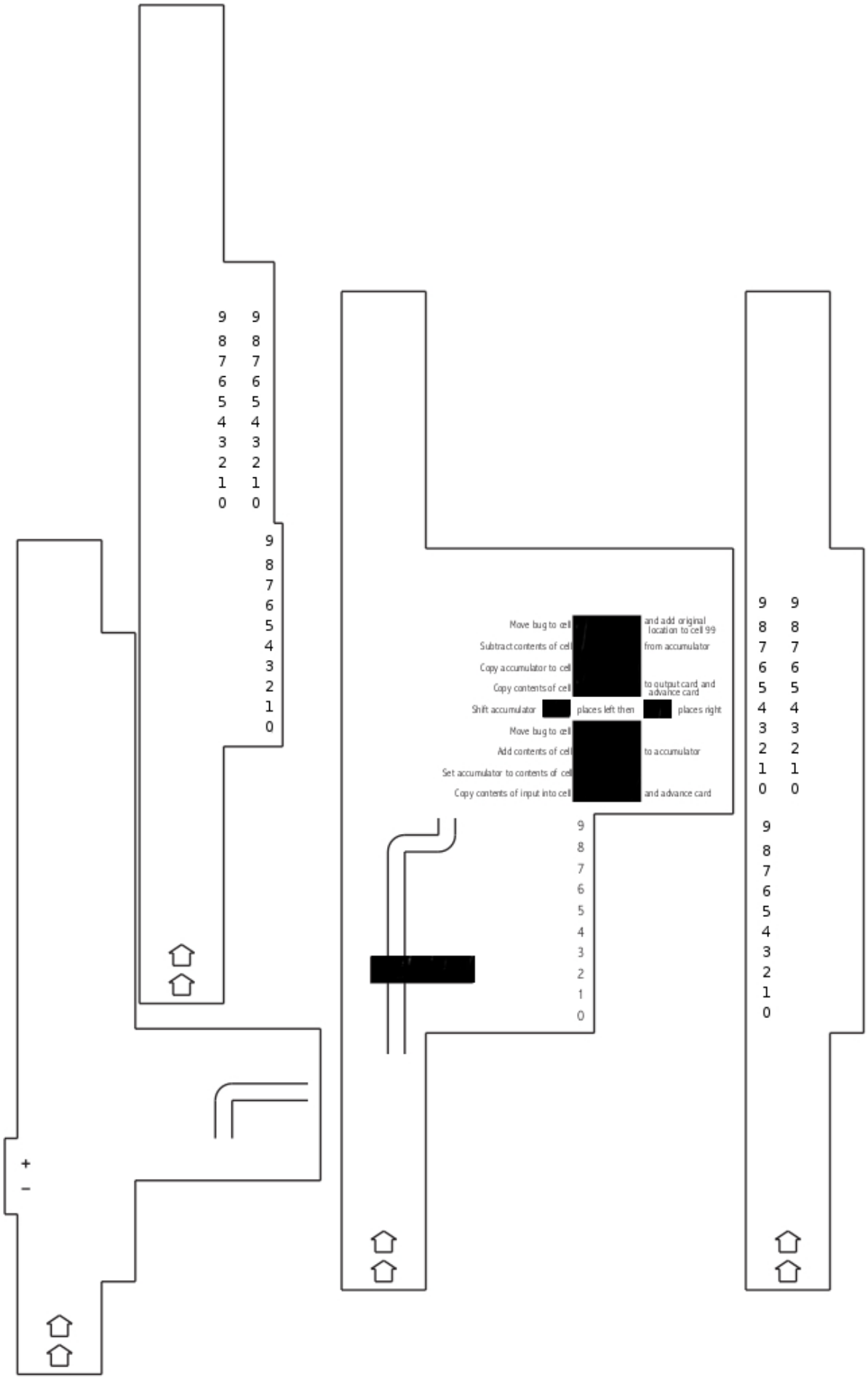
High level languages have never developed for CARDIAC, since they would defeat one of the purposes of the device: to introduce concepts of assembly language programming.

Programs are hand assembled then are penciled into the appropriate memory cells.

Operation

Programs are run by first sliding three slides so that the number in the instruction register equals the number in the memory cell the bug is sitting in. Once that was done the bug is moved to the next memory cell. The user then follows an arrow which will then tell them what to do next. This continues for all of program execution.

For more information please look on Facebook or Google+ for our CARDIAC Group or https://en.wikipedia.org/wiki/CARDboard_Illustrative_Aid_to_Computation


9 9
8 8
7 7
6 6
5 5
4 4
3 3
2 2
1 1
0 0
9
8
7
6
5
4
3
2
1
0
Move bug to cell
and add original location to cell 99
Subtract contents of cell
from accumulator
Copy accumulator to cell
Copy contents of cell
to output card and advance card
Shift accumulator
places left then
places right
Move bug to cell
Add contents of cell
to accumulator
Set accumulator to contents of cell
Copy contents of input into cell
and advance card
9 9
8 8
7 7
6 6
5 5
4 4
3 3
2 2
1 1
0 0
9
8
7
6
5
4
3
2
1
0
+
–